

ЯКІСТЬ ВИХІДНОГО КОДУ У ЗВ'ЯЗУ З ПІДТВЕРДЖЕНОЮ ТЕХНІЧНОЮ ЗАБОРГОВАНІСТЮ

Гринько А. М.

Науковий керівник – д.т.н, проф. Смеляков К.С.

Харківський національний університет радіоелектроніки
(61166, Харків, просп. Науки, 14, каф. Програмної інженерії,
тел. (057) 702-14-46), e-mail: alina.hrynko@nure.ua

The purpose of this study is to find a connection between code quality issues, found by SonarQube and those marked as SATD. The obtained result demonstrates that the introduction of SATD is usually related to certain types of issues.

Важливість якості програмного коду швидко зростає. Оскільки щодня пишеться більше коду, його обслуговування та підтримка стають все складнішими та дорожчими. Нові засоби автоматичного перегляду коду розроблені для досягнення кращої якості. Одним з таких інструментів є SonarQube.

Незважаючи на численні дослідження якості програмного забезпечення, розробники все ще пишуть недосконалий код, який потребує переробки в майбутньому, або спричиняє проблеми. Деякі приклади включають неправильний вибір структури коду, дублікати коду, жорстко закодовані параметри тощо. Це зазвичай робиться для того, щоб пришвидшити процес розробки, укластися в терміни або зменшити витрати. Сама це називається «технічним боргом». Ця метафора була вперше представлена В. Каннінґемом у 1994 р. [1] і використана для інкапсуляції численних проблем якості програмного забезпечення, що є досить поширеним явищем. Введення TD загалом означає, що розробник знижує якість вихідного коду, роблячи завдання виявлення та виправлення початкової проблеми складнішим. Незважаючи на те, що ці практики очевидно погані [2], технічний борг може бути частково виправданий прискоренням розробки й найшвидшим отриманням результату.

Аналогічно “боргу” в економіці, технічний борг може допомогти досягти деяких короткотермінових цілей, але його слід повернути (недосконалий код слід переробити) якомога швидше. Несплата технічної заборгованості може призвести до збільшення витрат у майбутньому. Також може бути набагато складніше завершити рефакторинг на пізнішому етапі, а також знайти непередбачувану помилку у старому коді. Існує декілька типів проблем, які ідентифікують потенційні вразливості архітектури, наприклад, «дубльований код» тощо.

Ситуацію, коли розробники чітко усвідомлюють, що вони «беруть технічну заборгованість» і згадують про неї, розкривають Потдар А. та Шихаб Е. у своєму дослідженні [2]. Вони запропонували термін SATD

(self-admitted technical debt), який загалом стосується ситуації, коли розробник вводить код із коментарем, наприклад "ToDo: Виправте це пізніше» або залишає примітку в будь-якому іншому каналі зв'язку (наприклад, квитки на Jira). Одним з найбільш поширених методів знаходження SATD у текстових коментарях подібного типу є алгоритми обробки природної мови (NLP) [3].

Метою цього дослідження є пошук зв'язку між проблемами якості коду, виявленими SonarQube, та проблемами, позначеними як SATD.

В результаті досліджень було виявлено певний відсоток SATD. Він становить від 0% до 20,83%. Не було виявлено зв'язку між розміром проекту та відсотком SATD. Є певні проблеми, які, мають певне відношення SATD, такі як «дубльований код», «невикористані параметри методу слід видалити», «когнітивна складність методів не повинна бути занадто високою» тощо. Введення SATD має незначний позитивний вплив на час виправлення помилок.

Ми сподіваємось, що наші висновки можуть допомогти покращити підходи до оцінки якості коду та політику розробки.

Перелік використаних джерел:

1. Cunningham W. The WyCash portfolio management system //ACM SIGPLAN OOPS Messenger. – 1992. – Т. 4. – №. 2. – Р. 29-30.
2. Potdar A., Shihab E. An exploratory study on self-admitted technical debt //2014 IEEE International Conference on Software Maintenance and Evolution. – IEEE, 2014. – Р. 91-100.
3. Smelyakov K., Chupryna A., Karachevtsev D., Kulemza D., Samoilenko Y., Patlan O. Effectiveness of Preprocessing Algorithms for Natural Language Processing Applications // 2020 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), 6-9 Oct. 2020, Kharkiv, Ukraine. – Р. 1-5.